Machine Learning for Detecting Credit Card Frauds

Atika Gupta, Bhaskar Pant, Nidhi Mehra, Divya Kapil

Abstract: ***Credit card frauds has been a threat that has evolved as a major source of loss for the financial sectors. It has been seen in the different parts of world causing loss of billions of dollars. It is also a area which needs attention from the researchers as the task of fraud detection can be automated using the different machine learning classifiers and data science. If the frauds model encounter the fraudulent transactions it will raise an alarm to the system administrator. The paper proposes a model which uses the machine learning classifiers to detect the fraudulent transactions. The classifiers used in the paper are SVM (Support Vectore Machine), Isolation Forest and Local Outlier. The focus of the research is to detect the fraudulent transactions to 100% and also we emphasise on the fact that no normal transaction should be detected as fraud wrongly. The process starts with preprocessing the data and then the classifers are applied. The results from each classifers is evaluated to check the one with the better performance. The performance can be increased with use of deep learning algorithms but with the rise in expennses.***

Keywords: credit card fraud, machine learning,isolation forest, local outlier.

I. INTRODUCTION

Credit Card frauds is a severe problem which any card issuer may face. Discussing about the losses, USA has faced a loss of 800 million dollars in the year 2004 due to credit card transactions, on the other side the same year UK has faced a loss of 425 million pounds due to these transactions.[1] Taking the discussion forward the losses incurred due to credit card frauds reaches 3 billion dollars in 2017 in North America. The meanas which were used are the Apple Pay, Venmo and Android Pay.[2]

In formal terms credit card fraud can be described as "*When a persona tries to use the credit card of another individual for his/her personal use when the credit card holder is unaware of the theft. To add, the user does not have any connection with the card issuer and neither aims to establish one*". The frauds occurance forms can take one of the several forms like: Lost card – 48%, Identity theft – 15%, Cloning – 14%, Counterfeit – 12%, Mail – 6% and other – 5%. There are many ways in which the attacker or the thieve is executing the fraud. [3] This is a problem which is a mattern of concern and needs attention from the researchers where AI can help in automating the solution to the problem. The figure 1 given below describes the frauds and the occurrence percentage.

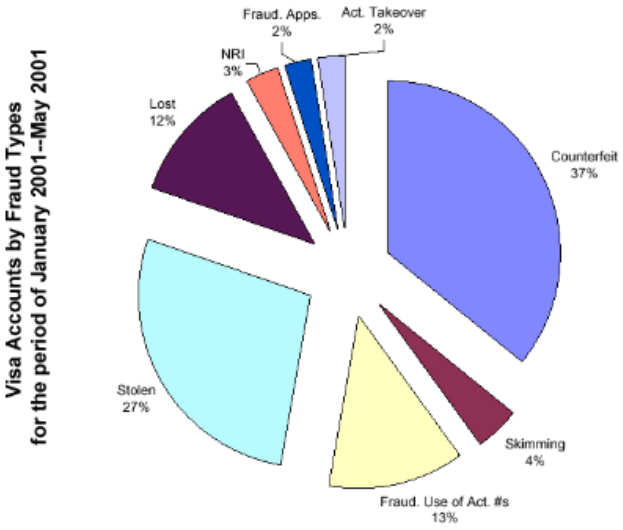

Figure1:Fraud Account and types

The paper has discussed the use of AI in detecting these credit card fruads. The branches of AI, Machine Learning and Deep Learning has proved to be promising in these detections. Several algorithms of Machine Learning such as Local Outlier Factor, Isolation Forest, SVM(Support Vector Machine), Logistic Regression, Decision Tree, Random Forest can be applied for the same purpose[4]. The major problem which is seen here is the class imbalance where the number of fradualent transactions are far less than the number of normal transactions.[5]

In this paper, our contribution is divided into the following sections. The discussion in Section II is about related Work which is done about the topic. Section III tells the Fraud Scenario using ML, section IV discusses the methodology in which we describe the dataset and the different algorithms which are applied on that dataset, section V show the implemention, how the implementation has been performed, section VI defines the reults obtained and discussion, And finally, the conclusion our paper in section VII.

II. RELATED WORK

[6]With the increase of uses of internet and credit crads, online shoppings and banking have grown tremendously. With the growth of all these new era technologies, credit card frauds have also growing. There are many modern day techniques evolved in detecting these fradulant transactions such as AI and Datamining. The article proposes a model which uses decision tree algorithm for fraud detection. These are used in combination of Luhn's and Hunt's algorithm which performs very well using the outliers.

Revised Manuscript Received on September 25, 2019.

Atika Gupta, School of Computing, Graphic Era Hill University, Uttarakhand, India. E-mail: atika04591@gmail.com

Dr. Bhaskar Pant, Department of CSE, Graphic Era Deemed to be University, Uttarakhand, India. E-mail: pantbhaskar2@gmail.com

Nidhi Mehra, School of Computing, Graphic Era Hill University, Uttarakhand, India. E-mail: nidhigehu@gmail.com

Divya Kapil, School of Computing, Graphic Era Hill University, Uttarakhand, India. E-mail: divya.k.rksh@gmail.com

[7]The article aims to focus on improved models of credit card fraud detection which are based on Machine learning and Nature Inspired detection techiniques. The Machine Learning models which are used are: NN(Neural Networks), HMM(Hidden Markov Model), SVM(Support Vectore Machine), and Decision Tee, whereas the Nature Inspired models includes: Genetic Algos and Artificial Immune System

[8]The discussion here is to deal with credit card fraud detecting model which is based on SVM(Support Vector Machine). The the dataset was very large to handle so PCA(Principal Component Analysis) is used to wipe out the attributes which are not required or are imbalanced for SVM to handle. With the real time dataset used it is concluded that ICG-SVM(Imbalanced Class Weightage SVM) is able to handle the fraud detection with high-precision.

[9]Credit card fradulant transactions cause an annual los of billion of money to the industries. The papers approach is to use the aggregation strategy to detect the frauds. Customer buying patterns are analysed with the help of aggregated transations and then the aggregated models are used for identifying the frauds.One of the focus area for the study is to highlight the partition area of the dataset where the threats are more likely to occur which is important because it is very difficult to check on all the transactions of the given dataset given the constraint of time and cost.

[10]The main idea of this paper is to focus on real-time detection of fraud and give a naïve and innovative idea of understanding customer spending patterns to detect the fradulnt transactions.Self-organising maps are used to decipher, analyse and filter the behaviour of the customer in fraud detection. The study has a focus on filtering and clustering qualities in fraud detection. Clustering is used to identify the patterns which are hidden and filtering on thte other hand is used for reducing the transaction size in order to save cost and processing time.

[11] The paper uses Hidden Markov Model for credit card transaction processing and gives an idea about how it can be used to detect the fraud. HMM model is first set to training with normal consumer behaviour and after that it is tested. If the incoming transaction is not accepted by the trained HMM model that means the transaction is fraudulent and it cancelled. In the process the model also makes sure that no normal transaction is rejected in suspicion of fradulaent one. It is observed that the model presents an occuracy of about 80%.

[12]The presentation here is the use of neural data mining. The concept which is developed here is the statistics based fraud detection where it is shown that the task is based upon very small proportion of fraud. It is also observed that by using the algorithmic generation of transaction data, a large number of diagnostic rules can be generated. By using the automaticaaly generated rules the correct diagonis has been incresead significantly.

[13]Due to the high losses which are happening because of credit cards it is becoming a serious matter for concern. The paper presented the use of two machine learning algorithms for classification which are: artificial neural networks and Bayseian algorithm. The study shows that the Bayseian model yields better results in contet of credit card frauds and the training time is usually shorter, the when the speed is considered, ANN is much faster.

[14]The paper states the problem of preprocessing the data from transaction for fraud classification. It is not practical to present all the transactions into the model not only because of the high dimentionality but also because the data is heterogenous. Therefore a framework which considers the transaction aggregation is implemented and its effectiveness is evaluated. Although transaction aggregation is found to be effective in many situations but not all situations fit. It performs the best when the random classifier is used.

[15]In the paper, the patterns and the characteristics are checked with the help of data processing to check the normal and the fraudulent transaction. At the same time, machine learning classifiers are used to predict the transaction based on victimization. The data was being preprocessed with the help of PCA and normalization. All the classifiers gave an accuracy of 95% compared to unprocessed data.

III. FRAUD SCENARIO USING ML

Fraud is considered as an act of illegally depriving a persons account for money or any legal rights. These frauds are so frequent as a company loses almost 5% of its revenue due to fraud[16].

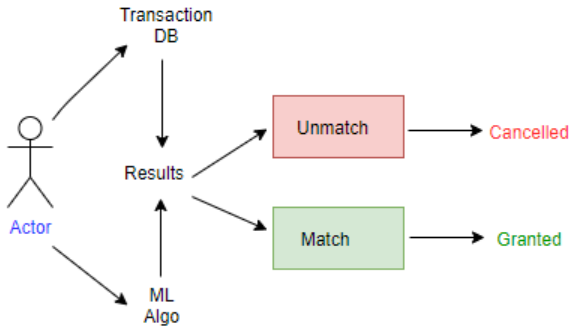

Figure 2: Fraud Scenario

The fraud scenario explains how actually the frauds can be controlled with the use of ML. There is an actor/customer who fires the transaction, the transaction is sent to the transaction DB for the results and at the same time it is fed to the machine learning model for detection. If the results from both the engines match that means that the transaction is real or normal and it is granted. If the transactions from both the engines does not match that means that the transaction is fradulant and it is cancelled. That is how we can control these increasing monetary frauds.

We also show here how the machine learning models are applied to create a training and testing model for fraud detection[17].

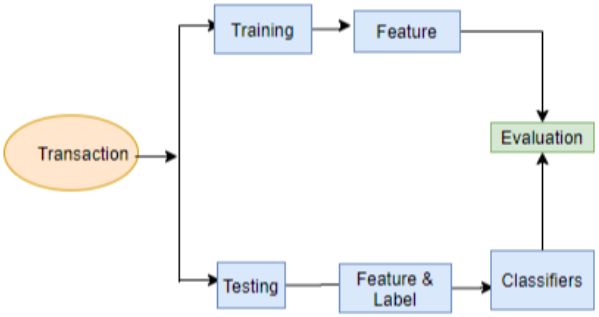

Figure 3: Training Model

The figure explains how we are building the training and testing model. The transactions are fed to the training as well the testing model where training model has the feature selection and classifiers are applied to the testing model and then the final evaluation is done to check the efficiency of the training model.

IV. METHODOLOGY

For the experiment we have used CreditCard datset which is imported from Kaggle website, a website which provides datasets. The details of the dataset are given in the table below:

DATASETS AND THEIR DETAILS

Dataset	Samples	Features	Normal	Fraud
CreditCard	284807	31	284315	492

The dataset consist of a total of 284807 samples and a total of 31 features from which 284315 samples of credit card transactions are found to be normal whereas 492 samples are found to be fraudulent. Out of total 31 attributes, 28 are named as V1-V28 for security purpose and the rest are Time, Amount and Class. Here, Time represents the time elapsed between one transaction and other, Amount represents the amount transacted and Class denotes the transaction type where there are two types of values: 0(normal) and 1(fraudulent). It should also be noted that for the processing speed and faster testing 30000 smples of the dataset is used. Distibution is shown below:

Distribuition of Normal(0) and Frauds(1):

0 29904

1 94

In the results section we also present the results from the complete dataset as well with a total of 284807 samples.

The figure shown below shows a classification of normal and fradulant transaction in the form of bar chart.

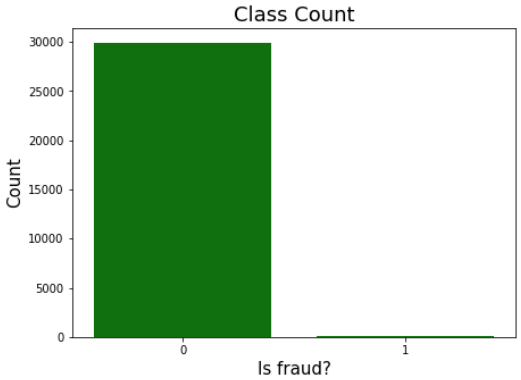

Figure 4: Fraudualent & Normal transaction

We have also classified the dataset according to the time distribution and it is shown in the figure below:

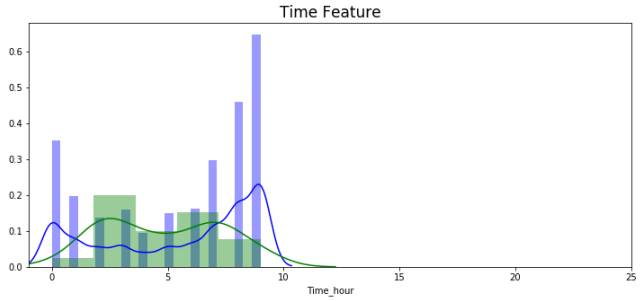

Figure 5: Time Feature Graph

The figure shows two bars, green and blue where the green one is the fraudulant and the blue bar is the legitimate one. It also shows that the fradulant transactions are much lower than the legitimate one. Also analyse the figure below for a pictorial representation of fraud and normal transactions by minute and hour.

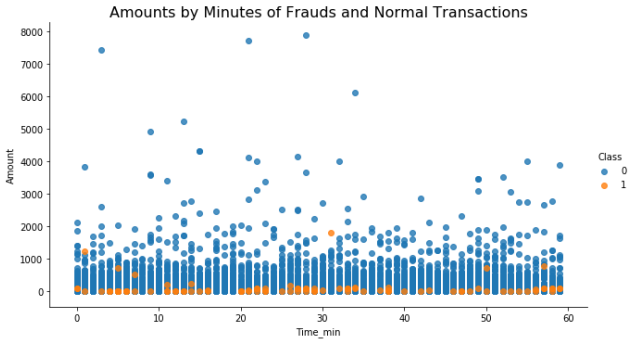

Figure 6: Minute Time Graph

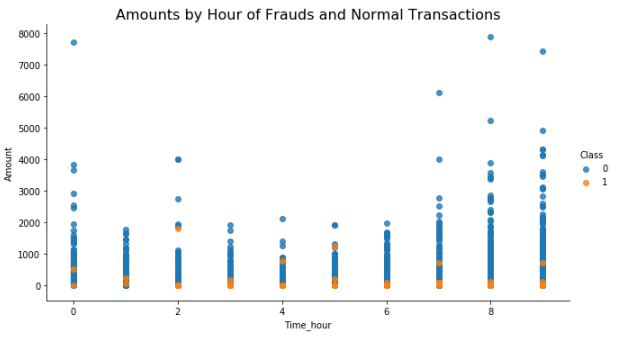

Figure 7: Hour Time graph

It will amaze you to know that the transaction which are fradulant are having lower amount as compared to normal transactions. Only a few of them appear closer to the maximum transacted amount. The graph below shows the same

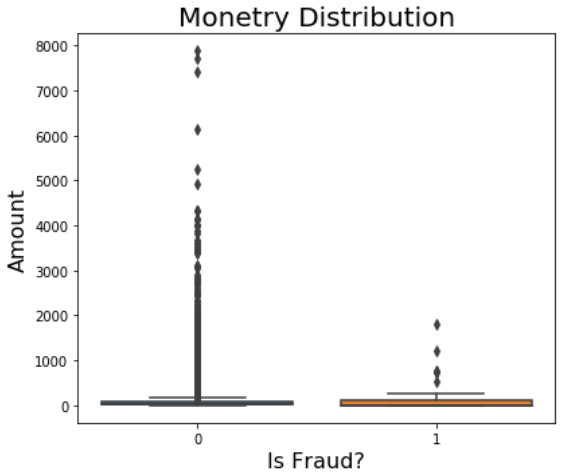

Figure 8: Monetry Distribution

Further a heat map which is a two-dimentional representation where the values are represented in colors is shown for better visualization of data.[18] It is also used to analyse and study the correlation in between the class and the predicting variables. The heatmap is shown below:

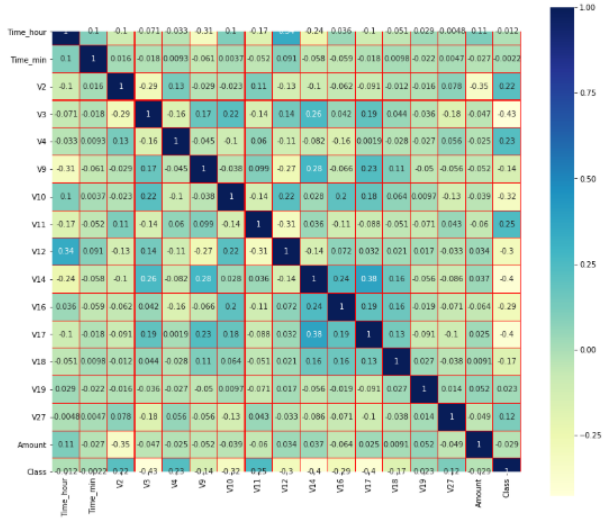

Figure 9: Heatmap

Moving forward we study that now the data has been preprocessed and the formatting is done. The time, amount and class colmns are analyzed and standardised in order to present a fair evaluation. The dataset is fed into some set of algorithms and is explained how these models work when brought together[19]. The two major algorithms which are used here are :

- Isolation Forest Algo
- Local Outlier Factor
- Support Vector Machine

The algorithms are being implemented on the dataset using Python. Sklearn is the package which contains these algorithms. The module in sklearn which contains the algos is ensemble module and contains the functions and methods for classification, regression and the detection of outlier.

The library is free and open-source and is built using the matplotlib, NumPy(Python Numerical) and SciPy (Scientific Libraries) modules which is considered as a very easy to use and efficient tool for analysis of data and machine learning. Sklearn has various algorithms like SVM(Support Vector Machine), random forest and K-neighbour algorithm.

Jupyter Notebook platform is used to illustrate the demonstration which has been carried out in the paper. Jupyter Notebook can also be accessed from Google Colab platform which is able to handle all the python files.

A. Isolation Forest Algorithm

It is the newest technique which is used in detecting anamolies. The basis of the algorithm is that the anamolies are some data points which are different and limited[20]. These properties and anamolies are suspected to be isolated. It uses isolation in an efficient and effective manner in place of the earlier used distance and density model. Moreover, the algorithm proves itself in the time and space complexity tradeoff[21].

Unlike the traditional machine leaning models which ised the balanced dataset for better predictions, Isolation forest works by isolating a randomly selected feature and then splitting at a randomly selected value which lies between the maximum and the minimum value of the selected feature. It is easy to do it as isolation only requires separating the values which shows different behaviour from the normal ones[22].

The algorithm proceeds by creating an Isolation tree or the random tree and then the score is calculated. The code below can be analysed for the same:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest

rng = np.random.RandomState(42)

# Generate train data
X = 0.3 * rng.randn(100, 2)
X_train = np.r_[X + 2, X - 2]
X = 0.3 * rng.randn(20, 2)
X_test = np.r_[X + 2, X - 2]
X_outliers = rng.uniform(low=-4, high=4, size=(20, 2))
```

```
# fit the model
clf = IsolationForest(max_samples=100, random_state=rng)
clf.fit(X_train)
y_pred_train = clf.predict(X_train)
y_pred_test = clf.predict(X_test)
y_pred_outliers = clf.predict(X_outliers)

# plot the line, the samples, and the nearest vectors to the plane
xx, yy = np.meshgrid(np.linspace(-5, 5, 50), np.linspace(-5, 5, 50))
Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
```

The plotting of the graph is shown below:

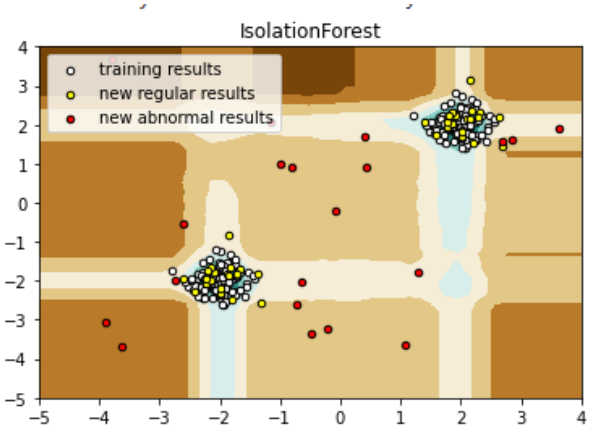

Figure 10: Isolation Forest

B. Local Outlier Factor

Unsupervised learning is a category of machine learning where the machine learns by itself. There are no pre-fed patterns as supervised learning and there is no or minimal human intervention. Local Outlier Factor falls in the category of unsupervised learning approach where the aim is to identify the local deviation of a given data point from its neighbours[23].

An outlier in broad terms is a data data point which is different from the others and a different mechanism is used to generate it. It is gaining popularity in fraud detection, criminal activity identification, and many others[24]. The neighbour parameter is chosen keeping in mind the two factors: 1) thte cluster has to contain a minimum number of object which is greater so that the other objects can have local outlier related to this cluster. 2) and it should be smaller than the maximum objects which are close and has a potential to become a local outlier.

Although there are rare events of outlier occurrence, in many examples which uses this methods such as online frauds and video surveillance for example, the importance which they hold is pretty high which makes its detection very important[25]. The code below can be analysed:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import LocalOutlierFactor

np.random.seed(42)

# Generate train data
X = 0.3 * np.random.randn(100, 2)
# Generate some abnormal novel observations
X_outliers = np.random.uniform(low=-4, high=4, size=(20, 2))
X = np.r_[X + 2, X - 2, X_outliers]
```

```
# fit the model
clf = LocalOutlierFactor(n_neighbors=20)
y_pred = clf.fit_predict(X)
y_pred_outliers = y_pred[200:]

# plot the level sets of the decision function
xx, yy = np.meshgrid(np.linspace(-5, 5, 50), np.linspace(-5, 5, 50))
Z = clf._decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
```

The plotting of the graph is shown below:

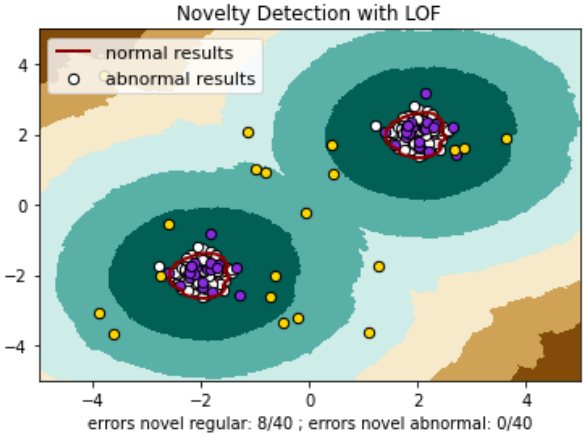

Figure 11: Local Outlier

C. SVM (Support Vector Machine)

SVM has emerged to be a new technique for classification and regression in machine learning. For a good generalization ability SVM uses a separating hyperplane for maximizing the margin. The application of SVM is dand-writing character recognisation, face detection, data mining etc[26]. SVM has its own limitations as it is a binary classifier and has to be combined with a muti-class classifiers for better results and secondly it is very time consuming. SVM ensemble are used to overcome these drawbacks of SVM algo.

To state in simple terms, SVM aims to find the best hyperplane which separates the two classes of samples from training set in feature space. A hyperplane is defined by a linear function f(x) = (y,m) + b, where the mapping of patterns is done in feature space y, and the value which defines f(x) is positive value[27].

The criteria which SVM uses is the margin maximization, where the word margin significe the distance which lies between the hyperplane of each class. Here we try to separate the classes in an hypothetical case where a perfectly separable class may never lie and the other class is the non-perfectly separable class which is also called the soft boundary where there are chances of errors and also they have to be minimized.[28]

SVM code below can be analysed :

```
##Define the outlier detection methods

classifiers = {
    "Isolation Forest":IsolationForest(n_estimators=100, max_samples=len(X),
                                       contamination=outlier_fraction,random_state=state, verbose=0),
    "Local Outlier Factor":LocalOutlierFactor(n_neighbors=20, algorithm='auto',
                                              leaf_size=30, metric='minkowski',
                                              p=2, metric_params=None, contamination=outlier_fraction),
    "Support Vector Machine":OneClassSVM(kernel='rbf', degree=3, gamma=0.1,nu=0.05, max_iter=-1)

}
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.datasets import make_blobs

# we create 40 separable points
X, y = make_blobs(n_samples=40, centers=2, random_state=6)
clf = svm.SVC(kernel='linear', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.BrBG)

# plot the decision function
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

# create grid to evaluate model
xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
Z = clf.decision_function(xy).reshape(XX.shape)
```

The plotting of the graph using SVM is shown below:

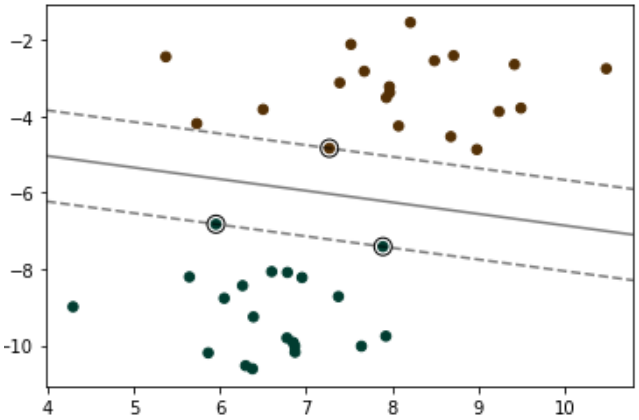

Figure 12: SVM Plot

V. IMPLEMENTATION

The implementation of the dataset is done using Python as previously discussed. The main consideration here is the dataset which is available, as it is not a good dataset because it contains more number of normal transactions ad only a handful of fradulaent one which makes it difficult to analyse the frauds detection system. One of the reason for this typ of dataset is confidentiality and privacy of the bank customers. Moreover the dataset is imbalanced and the features are misclassified and the algos of machine learning are not designed to handle such type of data.[7]

The code above can be analysed for the algorithms which are applied on the dataset.

VI. RESULTS AND DISCUSSIONS

```
n_outliers = len(Fraud)
for i, (clf_name,clf) in enumerate(classifiers.items()):
    #Fit the data and tag outliers
    if clf_name == "Local Outlier Factor":
        y_pred = clf.fit_predict(X)
        scores_prediction = clf.negative_outlier_factor_
    elif clf_name == "Support Vector Machine":
        clf.fit(X)
        y_pred = clf.predict(X)
    else:
        clf.fit(X)
        scores_prediction = clf.decision_function(X)
        y_pred = clf.predict(X)
    y_pred[y_pred == 1] = 0
    y_pred[y_pred == -1] = 1
    n_errors = (y_pred != Y).sum()
    # Run Classification Metrics
    print("{}: {}".format(clf_name,n_errors))
    print("Accuracy Score :")
    print(accuracy_score(Y,y_pred))
    print("Classification Report :")
    print(classification_report(Y,y_pred))
```

The code prints out the result using the three machine learning algos. The accuracy and precision score is also calculated. We have presented two results, one with 30000 samples and the other with the complete dataset.

The results are given not only in terms fof accuracy and precision but the complete classification is shown in the result.

Results with limited samples:

Support Vector Machine: 1426
Accuracy Score : 0.5246666666666666
Classification Report :

	precision	recall	f1-score	support
0	1.00	0.52	0.69	2992
1	0.00	0.50	0.01	8
accuracy			0.52	3000
macro avg	0.50	0.51	0.35	3000
weighted avg	0.99	0.52	0.69	3000

Isolation Forest: 7
Accuracy Score : 0.9976666666666667
Classification Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2992
1	0.56	0.62	0.59	8
accuracy			1.00	3000
macro avg	0.78	0.81	0.79	3000
weighted avg	1.00	1.00	1.00	3000

Local Outlier Factor: 17
Accuracy Score : 0.9943333333333333
Classification Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2992
1	0.00	0.00	0.00	8
accuracy			0.99	3000
macro avg	0.50	0.50	0.50	3000
weighted avg	0.99	0.99	0.99	3000

Results with complete dataset:

Support Vector Machine: 18113
Accuracy Score : 0.36403216179207193
Classification Report :

	precision	recall	f1-score	support
0	1.00	0.36	0.53	28432
1	0.00	0.63	0.00	49
accuracy			0.36	28481
macro avg	0.50	0.50	0.27	28481
weighted avg	1.00	0.36	0.53	28481

Isolation Forest: 73
Accuracy Score : 0.9974368877497279
Classification Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28432
1	0.26	0.27	0.26	49
accuracy			1.00	28481
macro avg	0.63	0.63	0.63	28481
weighted avg	1.00	1.00	1.00	28481

Local Outlier Factor: 99
Accuracy Score : 0.9965239984551104
Classification Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28432
1	0.00	0.00	0.00	49
accuracy			1.00	28481
macro avg	0.50	0.50	0.50	28481
weighted avg	1.00	1.00	1.00	28481

VII. CONCLUSION

Online transactions have become a need of the new era, since it requires only the authorization to use money for different puposes. But with the advantages which it provides it is also a medium for online crime. Many cases of credit card frauds have been detected causing major losses. The paper proposes models to detect online frauds, the models are machine learning based namely: isolation forest, local outlier and SVM. It is observed that the accuracy prediction with isolation forest is 99.74%, with local outlier is 99.65% and with SVM it is 70.07%. Further it should be considered that the dataset is imbalanced, for the future scope of the work, a more refined and balanced datset can be used.

REFERENCES

1. S. Aihua, T. Rencheng, and D. Yaochen, "Application of classification models on credit card fraud detection," *Proc. - ICSSSM'07 2007 Int. Conf. Serv. Syst. Serv. Manag.*, no. 1997, pp. 2–5, 2007.
2. A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," *2018 Syst. Inf. Eng. Des. Symp. SIEDS 2018*, pp. 129–134, 2018.
3. T. P. Bhatla, V. Prabhu, and A. Dua, "Understanding Credit Card Frauds," *Cards Bus. Rev.*, vol. 1, no. 6, pp. 1–15, 2003.
4. S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," *Proc. Hawaii Int. Conf. Syst. Sci.*, vol. 3, pp. 621–630, 1994.
5. I. Sadgali, N. Sael, and F. Benabbou, "Performance of machine learning techniques in the detection of financial frauds," *Procedia Comput. Sci.*, vol. 148, no. Icds 2018, pp. 45–54, 2019.
6. P. Save, P. Tiwarekar, K. N., and N. Mahyavanshi, "A Novel Idea for Credit Card Fraud Detection using Decision Tree," *Int. J. Comput. Appl.*, vol. 161, no. 13, pp. 6–9, 2017.
7. A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, pp. 937–953, 2017.
8. Q. Lu and C. Ju, "Research on credit card fraud detection model based on class weighted support vector machine," *J. Converg. Inf. Technol.*, vol. 6, no. 1, pp. 62–68, 2011.
9. S. Jha, M. Guillen, and J. Christopher Westland, "Employing transaction aggregation strategy to detect credit card fraud," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12650–12657, 2012.

10. J. T. S. Quah and M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1721–1732, 2008.
11. A. Srivastava, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection using Hidden Markov Model," *IEEE Trans. Dependable Secur. Comput.*, vol. 5, no. 1, pp. 37–48, 2008.
12. R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," *Proc. Int. Conf. Tools with Artif. Intell.*, pp. 103–106, 1999.
13. Ravelin, "Machine learning for fraud detection," no. May, 2002.
14. C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Min. Knowl. Discov.*, vol. 18, no. 1, pp. 30–55, 2009.
15. H. Ali Shukur and S. Kurnaz, "International Journal of Computer Science and Mobile Computing Credit Card Fraud Detection using Machine Learning Methodology," *Int. J. Comput. Sci. Mob. Comput.*, vol. 8, no. 3, pp. 257–260, 2019.
16. O. S. Yee, S. Sagadevan, and N. H. A. H. Malim, "Credit card fraud detection using machine learning as data mining technique," *J. Telecommun. Electron. Comput. Eng.*, vol. 10, no. 1–4, pp. 23–27, 2018.
17. A. Pumsirirat and L. Yan, "Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 1, pp. 18–25, 2018.
18. V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," *Procedia Comput. Sci.*, vol. 165, pp. 631–641, 2019.
19. S P Maniraj, Aditya Saini, Shadab Ahmed, and Swarna Deep Sarkar, "Credit Card Fraud Detection using Machine Learning and Data Science," *Int. J. Eng. Res.*, vol. 08, no. 09, pp. 110–115, 2019.
20. F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 413–422, 2008.
21. L. Sun, S. Versteeg, S. Boztas, and A. Rao, "Detecting Anomalous User Behavior Using an Extended Isolation Forest Algorithm: An Enterprise Case Study," 2016.
22. S. Hariri, M. Carrasco Kind, and R. J. Brunner, "Extended Isolation Forest," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2019.
23. M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "OPTICS-OF: Identifying local outliers," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1704, pp. 262–270, 1999.
24. Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1641–1650, 2003.
25. D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," *Proc. 2007 IEEE Symp. Comput. Intell. Data Mining, CIDM 2007*, no. Cidm, pp. 504–515, 2007.
26. H. C. Kim, S. Pang, H. M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern Recognit.*, vol. 36, no. 12, pp. 2757–2767, 2003.
27. M. E. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Trans. Neural Networks*, vol. 17, no. 3, pp. 671–682, 2006.
28. L. Auria and R. A. Moro, "Support Vector Machines (SVM) as a Technique for Solvency Analysis," *SSRN Electron. J.*, no. August, 2011.